iSymphony: An Adaptive Interactive Orchestral Conducting System for Digital Audio and Video Streams

Eric Lee
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
eric@cs.rwth-aachen.de

Henning Kiel
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
kiel@cs.rwth-aachen.de

Saskia Dedenbach
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
dedenbach@cs.rwth-aachen.de

Ingo Grüll
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
gruell@cs.rwth-aachen.de

Thorsten Karrer
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
karrer@cs.rwth-aachen.de

Marius Wolf
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
wolf@cs.rwth-aachen.de

Jan Borchers
Media Computing Group
RWTH Aachen University
52056 Aachen, Germany
borchers@cs.rwth-aachen.de

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.
ACM 1-59593-298-4/06/0004.

Abstract

We present *iSymphony*, an interactive orchestral conducting system for digital audio and video that adaptively adjusts to the user's conducting style. Using a digital baton, users may control the tempo, volume, and instrument emphasis of a digital audio and video recording of an orchestra. The system adaptively recognizes three gesture profiles: the four-beat neutral-legato pattern, an up-down pattern, and random gestures. The system uses an audio time-stretching algorithm we developed that allows the playback speed of a digital audio recording to be arbitrarily adjusted without changing its pitch. *iSymphony* is an example of how computers can enable more people to experience an interaction style normally limited to a few people (conductors), and is installed as part of the *It's Artastic!* exhibit at the Betty Brinn Children's Museum in Milwaukee, USA.

Keywords

conducting; gestures; adaptive gesture recognition; music interfaces; exhibits.

ACM Classification Keywords

H.5.1: Multimedia Information Systems — augmented reality, audio output, evaluation/methodology; H.5.2: User Interfaces — evaluation/methodology, user-centered design.

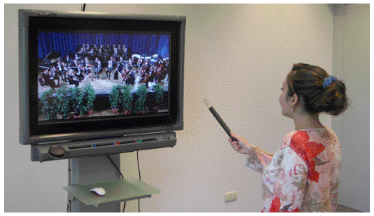

Figure 1. A user conducting an electronic orchestra using an *iSymphony* prototype.

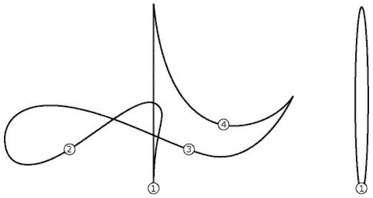

Figure 2. Baton trajectories for two of the three conducting styles that *iSymphony* recognizes. The beats for the four beat neutral legato pattern (left) are synchronized with the music at the numbered points. For the up-down style (right), the beat is synchronized to the lower turning point. The third conducting style assumes no regular pattern.

Introduction

Despite ongoing research in human-computer interaction, interfaces with multimedia remain relatively stagnant. Interaction with time-based media in particular, such as audio and video, remain largely limited to the decades-old "tape recorder" metaphors of play, stop, fast forward, and rewind.

Interactive conducting systems break this traditional metaphor by enabling users to interact with computer music using a digital baton. Most interactive conducting systems allow users to control, via gestures, the tempo, volume, and possibly instrument emphasis of an electronic orchestra. The temporal interaction of being able to arbitrarily change the playback speed of the orchestra is particularly memorable for users.

iSymphony improves the conducting interaction over our previous work, including *Personal Orchestra* [2] and *You're the Conductor* [5]. *Personal Orchestra* recognized only simple up-down conducting gestures, where the beat of the music is synchronized to the lower turning point of the baton. *You're the Conductor*, which was designed for children in collaboration with Teresa Marrin Nakra, did not require the user to conduct in a specific pattern. *iSymphony*, in contrast, allows users to conduct using one of three types of gestures, and adapts the system behavior accordingly, a characteristic that is unique to our system.

Interaction

As a user passes near the exhibit, she sees a large display roughly 2.3 meters wide. The display shows a looping video of an orchestra softly tuning their instruments — the audio is loud enough to attract attention, but soft enough to not disrupt. As she looks more closely, she sees a baton resting on a conducting podium in front of the screen. When she picks up the baton, the video transitions to a still picture of the orchestra holding their instruments at ready. As she begins waving the baton around, the video and music follow her gestures, speeding up as she speeds up, slowing down as she slows down; their volume also changes with the size of her gestures. When she gestures towards the violins, she hears them more distinctly above the rest of the orchestra (see Fig. 1).

She then hands the baton to her friend, who tries to conduct in a four-beat conducting pattern — the system recognizes his gestures, and in addition to letting him control the tempo, volume and instrument emphasis, the system synchronizes the beat of the music to his gestures. When he reaches the end of the piece, he is rewarded with applause, and the system returns to tuning their instruments, awaiting the next conductor.

Physical Design

The physical design of *iSymphony* was dictated by a series of *design patterns* for interaction [1]. The large, Immersive Display draws the user into the conducting experience, and attracts passer-bys to the exhibit. Most of the hardware used to run the exhibit is placed out of sight of users (Invisible Hardware), to hide the technical complexity from them. Finally, the digital baton is a Domain-Appropriate Device, a natural choice for an object to control music tempo and dynamics.

Software Design

The software that we designed for *iSymphony* is divided into two subsystems: gesture recognition/ interpretation, and audio/video rendering.

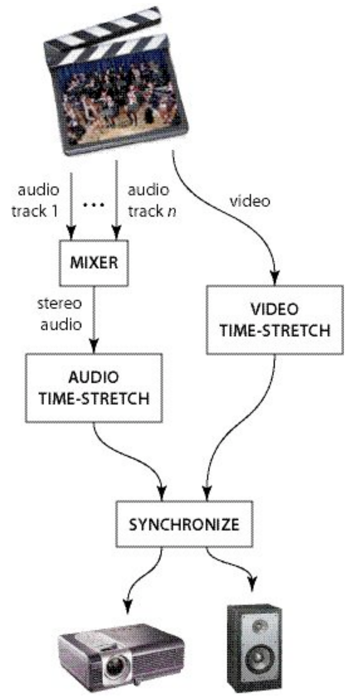

Figure 3. The Semantic Time Framework graph for the *iSymphony* rendering engine. The movie contains a video track and multiple audio tracks, one for each instrument section. The audio is down-mixed to a stereo track to reflect the instrument emphasis. Both the audio and video are then separately time-stretched and resynchronized before they are sent to the output devices for display and playback.

Gesture Recognition/Interpretation

We used a “feature detector” approach for tracking characteristics such as position, velocity, and acceleration of two-dimensional trajectories over time. As opposed to other gesture recognition approaches based on statistical methods or neural networks, our feature detector approach does not require the system to be trained for a particular user or set of users, which allows our system to be available to a wider audience. We designed and implemented a framework based on this feature detector approach: *conga* (Conducting Gesture Analysis Framework) [3]. A *conga* graph consists of linked processing nodes; these nodes detect features present in the gesture input and track the user's current beat in a conducting gesture.

We created *conga* graphs for each of the three gesture profiles that we support: the four-beat neutral-legato conducting pattern (see Fig. 2), up-down gestures where the beat is synchronized on the lower turning point, and random gestures where only the speed of the gestures is mapped to the music, but not the actual beat. The volume of the music is determined by the size of the gesture, and the instrument emphasis by the center of the gesture.

Audio/Video Rendering

We chose to use digital audio and video recordings for enhanced realism. We wanted to reproduce an authentic experience with a particular orchestra in a particular location, characteristics which are still not possible with today's synthesizing technology. However, continuously adjusting the temporal properties of digital audio and video recordings, where the semantics of the media are not explicitly known, resulted in two problems: how to arbitrarily alter the playback speed of the digital orchestral recordings in real-time, and developing a playback engine capable of presenting synchronous audio and video output in response to continuous user input.

The naïve method of time-stretching digital PCM audio by simply changing its playback speed results in undesirable pitch-shifting artifacts. *iSymphony* uses a variant of the phase vocoder algorithm with multiresolution peak-picking, a technique we designed to reduce some of the audio artifacts exhibited by existing audio time-stretching algorithms. Multiresolution peak-picking takes into account the non-linear frequency response of the human ear; details of the algorithm can be found in [4].

Modern, well-known multimedia frameworks seldom support custom effects that continuously adjust the temporal properties of the media (e.g., time-stretching). We designed a new multimedia framework with the concept of *semantic time* that addresses an issue of varying time models amongst differing media types [4]. Our Semantic Time Framework supports a graph-style architecture, where nodes that apply digital signal processing effects are applied to the audio and video (see Fig. 3). Time, in the form of semantic time units (beats), is preserved throughout this processing pipeline. Unlike the number of audio samples, the number of semantic time units does not change, even after time-stretching.

Implementation

The *iSymphony* software was implemented to run on Mac OS X. The Semantic Time Framework was written in C++, using QuickTime and Core Audio to render the processed video and audio. The rest of *iSymphony* was

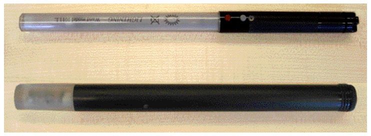

Figure 4. Buchla Lightning II digital batons. The baton on the top is the original baton, made from aluminum and fiberglass. The baton on the bottom is our repackaged version of the baton made from plastic and resin for better durability.

implemented in Objective-C as a native Mac OS X application.

The digital baton is a repackaged Buchla Lightning II system. The original metal and fiberglass baton packaging was exchanged in favor of a custom padded shell made from tough plastic and resin that is more robust to heavy use, especially by children (see Fig. 4). We also removed the buttons, which are unnecessary for *iSymphony*.

Our recordings were filmed during an exclusive recording session with a local student orchestra. A separate audio track was created for each of the seven instrument sections; these audio tracks are mixed down to a stereo track at runtime, taking into account the user's requested instrument emphasis. The system requires the position of the beats in the music, and this metadata is manually created offline using *BeatTapper*, a simple tool we wrote which allows someone to tap out beats while listening to the music, and then manually aligning them to the amplitude waveform of the music.

Conclusions

We presented *iSymphony*, an interactive conducting system that allows users to control the tempo, dynamics, and instrument emphasis of an electronic orchestra. Users may gesture with a digital baton in one of three ways: a four-beat neutral-legato conducting pattern, up-down movements, or no pattern at all. With the former two gestures styles, the beat of the music is synchronized to the users gestures in addition to the tempo. We use a "feature-detector" approach for the gesture recognition by building *conga*, a reusable framework for recognizing conducting gestures. The audio/video rendering engine of *iSymphony* uses an algorithm we developed for high-quality time-stretching of polyphonic digital PCM audio. This time-stretching module is part of the Semantic Time Framework, a software library we designed for supporting time-based interactions with multimedia. *iSymphony* is installed as part of the *It's Artastic!* exhibit at the Betty Brinn Children's Museum in Milwaukee, USA.

Acknowledgements

Personal Orchestra was coordinated by Max Mühlhäuser, now at Darmstadt University, and in cooperation with the Vienna Philharmonic. *You're the Conductor* was developed in cooperation with Teresa Marrin Nakra at Immersion Music, and the Boston Symphony Orchestra. *iSymphony* was sponsored by the Betty Brinn Children's Museum in Milwaukee, USA.

References

[1] Borchers, J. *A pattern approach to interaction design*. John Wiley & Sons, New York, 2001.

[2] Borchers, J., Lee, E., Samminger, W., and Mühlhäuser, M. Personal Orchestra: A real-time audio/video system for interactive conducting. *ACM Multimedia Journal Special Issue on Multimedia Software Engineering* 9(5), 2004, 458—465, Errata published in next issue.

[3] Grüll, I. conga: A conducting gesture analysis framework. Diploma thesis, University of Ulm, 2005.

[4] Lee, E., Karrer, T., and Borchers, J. Towards a framework for interactive systems to conduct digital audio and video streams. *Computer Music Journal*, 30(1), 2006, 21—36.

[5] Lee, E., Marrin Nakra, T., and Borchers, J. You're the Conductor: A realistic interactive conducting system for children. *Proc. of NIME 2004*. 68—73.